

Scenario Validation

This document validates the domain model by walking through six realistic recruiter–candidate interaction scenarios. Each scenario demonstrates whether the current design supports the flow without inconsistencies or awkward workarounds.

Scenario 1: Recruiter Invites Candidate to an Event

Steps:

1. RecruiterProfile publishes an upcoming recruiting event.
2. Candidate sees the event in the application.
3. Recruiter sends an Invite; system checks candidate exists and event has capacity.
4. Candidate accepts.
5. Participation record is created linking Candidate ↔ Event.

Validation:

- Supported by existing entities (RecruiterProfile, CandidateProfile, Event).
- Association handled through Participation.
- No awkward workaround needed.

Scenario 2: Candidate and Recruiter Mutually Swipe

Steps:

1. Candidate swipes right on RecruiterProfile.
2. Recruiter swipes right on CandidateProfile.
3. A Match entity is created.
4. Messaging is unlocked between Candidate ↔ Recruiter.

Validation:

- Matches align with defined Swipe → Match → Connection flow.
- Message entity is dependent on a valid Match.
- Rules prevent duplicates.

Scenario 3: Candidate Passes on a Recruiter Profile

Steps:

1. Candidate views a RecruiterProfile card.
2. Candidate swipes left (Pass).
3. That profile is removed from the Candidate's discovery feed.
4. System records the Pass, preventing the same profile from reappearing in the session.

Validation:

- Supported by Swipe entity.
- Domain rules enforce Pass is final for the session.

Scenario 4: Recruiter Filters Candidates by Skill

Steps:

1. Recruiter opens candidate search feature.
2. Applies filters (e.g., Python skill + availability).
3. System returns CandidateProfiles meeting criteria.
4. Recruiter swipes right or left on filtered candidates.

Validation:

- Supported by Recruiter search and filter functions.
- Clean interaction between RecruiterProfile, CandidateProfile, and filter criteria.

Scenario 5: Candidate Updates Profile After a Match

Steps:

1. Candidate has an existing Match with a Recruiter.
2. Candidate updates StudentProfile (adds project/certificate).
3. Updated info becomes visible to Recruiter via Match.
4. Recruiter reviews updated CandidateProfile inside the connection.

Validation:

- Supported by Updated → Active life cycle transition.
- Recruiters tied by Match see the most recent data.

Scenario 6: Event Capacity Limit Reached

Steps:

1. Recruiter creates event with capacity (e.g., 50 attendees).
2. Candidates RSVP to the event.
3. When the 50th Candidate accepts, system blocks further RSVPs.
4. Candidate #51 is shown 'Event is full'.

Validation:

- Supported by Event entity and RSVP flow.
- Matches domain rule: system must enforce registration rules and capacity limits.